REMARKS

I. Notice of References Cited

The Examiner cited "Using the SNAP Development Environment," by Template Software and "Using the Web Component," by Template Software in the Final Office Action. The two references, however, do not appear on the Notice of References Cited. Accordingly, Applicants respectfully request that the Examiner list the two references and any other references published by the Template Software cited in the Final Office Action on the Notice of References Cited.

II. Regarding the Office Action

In the Final Office Action mailed February 23, 2007, the Examiner rejected claims 1-9 under 35 U.S.C. § 101 as being purportedly directed to non-statutory subject matter; rejected claims 1, 3-15, 17-18, and 20 under 35 U.S.C. § 102(b) as being unpatentable over SNAP ("Using the SNAP Development Environment," by Template Software) and WEB ("Using the Web Component," by Template Software); and rejected claims 2, 16, and 19 under 35 U.S.C. § 102(b) or 35 U.S.C. § 103(a) as being unpatentable over the commercial product line by Template Software in view of Development Tools. Based on the following remarks, Applicants respectfully traverse the rejections of the pending claims.

A. Rejections of Claims 1-9 Under 35 U.S.C. § 101

The Examiner asserted that claims 1-9 are directed to non-statutory subject matter because "[n]o physical transformation is recited and additionally, the final result of the claim is a converted program which is not a tangible result because claim does

not explicitly claim the result as being tangibly embodied on a computer readable." Office Action at 3. Although Applicants respectfully disagree, Applicants, in an effort to advance prosecution, propose to amend independent claim 1 to add a storing step, as suggested by the Examiner. Claims 2-9 depend from independent claim 1, and thus are directed to statutory subject matter by virtue of their dependence on independent claim 1. Accordingly, Applicants respectfully request reconsideration and withdrawal of the § 101 rejection of claims 1-9.

B. Rejections of Claims 1, 3-15, 17-18, and 20 Under 35 U.S.C. § 102(b)

The Examiner asserted that the SNAP programming language and the Web Component are "a single offering and constitute a single reference" because "[o]nce Web functionality is present and the Web component is installed the two are inseparable." Office Action at 10. Applicants respectfully submit that the Examiner's assertion is not supported by the references. First, nowhere does SNAP disclose anything related to the Web component, and thus a SNAP application or SNAP Development Environment is operable with or without the Web Component. Further, WEB discloses that the Web Component provides a "gateway, which manages the connection between end user web browsers and the requested, running SNAP application." WEB at 2-4. As shown in detail in Figure 2-1 of WEB, the gateway of the Web Component run **as a separate entity** between a HTTP server and a running SNAP application, or **as a plug-in attached to a HTTP server**. WEB at 2-4. Thus, the Examiner's assertion that the Web Component and SNAP application are "inseparable" is based on the Examiner's opinion and not supported by any disclosure made by the

references. Accordingly, Applicants respectfully request that the Examiner support the assertion based on the references before shifting the burden to Applicants to provide compelling case law that deals with dependencies of layered products.

Moreover, even if SNAP and WEB can be combined to constitute a single reference as asserted by the Examiner, the combination of SNAP and WEB fails to teach or suggest every claim element recited in independent claim 1. In particular, the combination of SNAP and WEB fails to teach or suggest "generating a **converted design-time representation of the application** based on the original design-time representation, the converted design-time representation for use in a second run-time environment for executing applications having been developed in a second design-time environment, the second design-time environment using a second programming model comprising one or more second model elements including models, views, and controllers, the converted design-time representation including one or more application views based on the one or more application screens, and **converted processing logic based on the original processing logic**, the converted processing logic capable of being executed in the second run-time environment." (emphasis added).

WEB discloses that the Web Component "converts . . . SNAP displays **at run time** to HTML or Java displays." WEB at 2-9. (emphasis added). Thus, the Web Component does not generate **a converted design-time representation** of SNAP displays. The Examiner asserted that "the format is not a binary 0's and 1's but HTML with embedded JAVA . . . stored in [files] . . . can be directly manipulated. Office Action at 11. Applicants respectfully disagree.

First, WEB teaches against storing a converted run-time representation of SNAP displays for later manipulation by disclosing that "(t)he SNAP display web pages **exist only as long as** they are accessed by end users." WEB at 2-4. (emphasis added). Thus, the converted run-time representation of SNAP displays exists only at run-time and is not stored as files for manipulation at design-time. Moreover, even if a converted run-time HTML representation of SNAP displays can be stored as files and manipulated later at design-time, a converted **run-time JAVA representation** of SNAP displays is binary (0's and 1's) and thus may not be manipulated later **at design time**. WEB fails to disclose that the Web Component enables a developer to manipulate a converted run-time JAVA representation of SNAP displays. Thus, the Web Component generates, at most, a converted run-time representation of SNAP displays, and fails to generate a converted design-time representation of SNAP displays.

Furthermore, the Web Component generates neither **a converted design time representation** nor **a converted run-time representation** of processing logic of a SNAP application. WEB discloses that converting SNAP application to run on the web does not involve "generating a converted **design-time representation** of the application based on the original design-time representation, . . . the converted design-time representation including . . . **converted processing logic** based on the original processing logic . . . " (emphasis added). Instead, the Web Component generates a converted **run-time representation** of **SNAP displays only** and creates a new object of the WEB ACCEPTER class, which allows a running SNAP application to accept connections from a HTTP server via the Web Component gateways. Id. Thus,

processing logic of a SNAP application runs "**unconverted**" and is only exposed to the web through the converted run-time representation of SNAP displays and the WEB ACCEPTER object. WEB at 2-10. In another words, rather than generating a converted design-time representation of processing logic of a SNAP application, the Web Component merely facilitates exposing the processing logic to the web.

This argument is further supported by WEB at 2-10 and 2-11, which details the steps involved in running a SNAP application on the web. At steps 3 and 4, the Web Component connects to a running SNAP application and "**converts the end-user request into an event or into data that the running SNAP application understands**. . . . The Web Component classes do this conversion . . . [and] **the conversion is transparent to the application's SNAP code**, as well." WEB at 2-10. Thus, rather than converting processing logic of a SNAP application, the Web Component converts end user requests for a SNAP application. Also, because the conversion of the end-user request is transparent to the application's SNAP code, the code does not need to know whether the request is originally from the web or from a standard SNAP client application.

At step 5, "[t]he **SNAP application processes the event or data**, and produces output." WEB at 2-11. (emphasis added). The "SNAP application executes with **little or no knowledge of its connection to the web** . . . [and] executes as though its end users were communicating with the application process directly [and not using the web]." Id. "While the **application generates displays in the same way as a regular SNAP application does**, you [i.e., developer] have developed the **displays for the**

application with the web-imposed features and constraints in mind." Thus, again, there are no changes made to processing logic of a SNAP application to run the application on the web. The processing logic remains "unconverted" within a SNAP application whether the application runs on the web or not, and processes the converted end user request without knowing that the request is from the web. The Web Component further puts the responsibility **on developers** to develop "web-aware" SNAP displays to enable a SNAP application to run on the web.

At step 6, "[t]he Web Component **converts the SNAP displays** generated by the [SNAP] application at run time into HTML and sends them to the Web Component gateway." WEB at 2-11.

For these additional reasons, converting a SNAP application to run on the web does not involve "generating a converted design-time representation of the application based on the original design-time representation, . . . the converted design-time representation including . . . converted processing logic based on the original processing logic . . . "

The Examiner asserted that WEB at 2-3 to 2-5 discloses "converted processing logic." However, WEB at 2-3 to 2-5 discusses generating a run-time representation of SNAP displays only.

> The Web Component enables you to develop SNAP applications whose **displays** are presented to end users as **web pages**. Through the Web Component software, **the SNAP application's displays** are **converted at run time** into **web pages** and **displayed via end users' web browsers**. When end users **access these web pages** through their browsers, they actually **interact with the SNAP application**. For SNAP applications, end users submit URL requests to connect to those applications in the same way that they request access to statically defined web pages;

> however, the URLs used to connect to SNAP applications specify the applications to be accessed and other necessary related information.
>
> The difference between simple, statically defined web pages and SNAP application displays presented as web pages is that **SNAP display web pages are dynamically generated** by an interactive program, ... The **SNAP display web pages exist only as long as they are accessed by end users**, while static web pages exist for some significant amount of time between manual updates.
>
> . . . In contrast, **displays used in SNAP applications** that run on the web are **HTML displays or Java applets** generated by the Web Component
>
> **Characteristics of the displays and applets** that run on the web are as follows:
>
> - **At run time**, the **Web Component converts** most kind of **SNAP application displays** to interactive **HTML documents**, and **converts other kinds of SNAP displays** (charts, canvases, and topologies) **to static Java applets**. . . .
> - SNAP applications on the web use sophisticated URL addressing schemes to prevent end users from accessing unauthorized portions of SNAP applications or other end users' application data.
> - **Certain features of SNAP do not operate over the web, because of the structure and limitations of HTML and of the various web browser programs**.

WEB at 2-3.

WEB at 2-4 discusses the structure of SNAP applications on the web: the end users' web browsers, a web server, the Web Component gateway, and a running SNAP application. WEB at 2-5 discusses the end user's web browsers, the web server, and the Web Component gateway in more detail. Therefore, nowhere does WEB at 2-3 to 2-5 teach "generating a converted design-time representation of the application based on the original design-time representation, . . . the converted design-time representation including . . . converted processing logic based on the original processing logic . . . "

For at least these reasons, SNAP and WEB, taken alone or in combination, fail to teach or suggest every claim element recited in independent claim 1. Independent claims 10, 15, and 18 recite features that are similar to the features recited in independent claim 1. Accordingly, Applicants respectfully request reconsideration and withdrawal of the § 102 rejection of independent claims 1, 10, 15, and 18 based on SNAP and WEB.

Claims 3-9 depend from claim 1, claims 11-14 depend from claim 10, claim 17 depends from claim 15, and claim 20 depends from claim 18. Thus, dependent claims 3-9, 11-14, 17, and 20 are allowable by virtue of their dependence on an allowable independent claim. Accordingly, Applicants respectfully request reconsideration and withdrawal of the § 102 rejection of dependent claims 3-9, 11-14, 17, and 20 based on SNAP and WEB.

C. Rejections of Claims 2, 16, and 19 Under 35 U.S.C. §§ 102(b) or 103(a)

The Examiner rejected claims 2, 16, and 19 under 35 U.S.C. § 102(b) or 103(a) "as being unpatentable over the commercial product line by Template Software in view of Development Tools." Office Action at 8. Claim 2 depends from claim 1, claim 16 depends from claim 15, and claim 19 depends from claim 18. Accordingly, claims 2, 16, and 19 are allowable at least by virtue of their dependence on allowable independent claims 1, 15, and 18, respectively.

With respect to the § 103 rejection, the Examiner asserted that "Template teaches the ability to build GUIs in **SNAP** to run as a local application and how to Web enable them with the product **WEB** which enables them to run in another environment."

Office Action at 8. Applicants respectfully submit that building a local application and "Web enabling" the local application is not the same as and does not necessarily involve "generating a converted **design-time representation of the application** based on the original design-time representation . . . " (emphasis added). As explained above with respect to independent claim 1, WEB teaches web enabling an application by converting an end-user request to an event or data that the application understands and converting output displays generated by the application at run-time to HTML pages without generating a converted design-time representation of the application. Therefore, the Examiner has not established *prima facie* case of obviousness with respect to claim 2, 16, and 19. Accordingly, Applicants respectfully request reconsideration and withdrawal of the §§ 102 or 103 rejection of claims 2, 16, and 19.

D. Conclusion

Applicants respectfully request that this Amendment under 37 C.F.R. § 1.116 be entered by the Examiner, placing claims 1-20 in condition for allowance. Applicants submit that the proposed amendment of claim 1 does not raise new issues or necessitate the undertaking of any additional search of the art by the Examiner, since all of the elements and their relationships claimed were either earlier claimed or inherent in the claims as examined. Therefore, this Amendment should allow for immediate action by the Examiner.

Furthermore, Applicants respectfully point out that the final action by the Examiner presented some new arguments as to the application of the art against Applicants' invention. It is respectfully submitted that the entering of the Amendment

would allow the Applicants to reply to the final rejections and place the application in condition for allowance.

Finally, Applicants submit that the entry of the amendment would place the application in better form for appeal, should the Examiner dispute the patentability of the pending claims.

In view of the foregoing remarks, Applicants submit that this claimed invention, as amended, is neither anticipated nor rendered obvious in view of the prior art references cited against this application. Applicants therefore request the entry of this Amendment, the Examiner's reconsideration and reexamination of the application, and the timely allowance of the pending claims.

Please grant any extensions of time required to enter this response and charge any additional required fees to Deposit Account No. 06-0916.

Respectfully submitted,

FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, L.L.P.

Dated: May 23, 2007

By: WJ Burgan Reg # 43,515
for Jeffrey A. Berkowitz
Reg. No. 36,743